

METAFONT, METAPOST and a Malayalam font

C.V. Radhakrishnan K.H. Hussain K.V. Rajeesh

July 15, 2023
TUG'23, Bonn

Rachana Institute of Typography

Malayalam script

900+, *shape-shifting* characters

- Malayalam script evolved from Brahmi, Grantha
- ~120 base characters encoded in Unicode
- ~800 conjuncts characters, formed from sequence of base characters

900+, *shape-shifting* characters

- Malayalam script evolved from Brahmi, Grantha
- ~120 base characters encoded in Unicode
- ~800 conjuncts characters, formed from sequence of base characters
- Shape shifting

900+, *shape-shifting* characters

- Malayalam script evolved from Brahmi, Grantha
- ~120 base characters encoded in Unicode
- ~800 conjuncts characters, formed from sequence of base characters
- Shape shifting

k+r+u: ക+ര+ു → ക്രു

900+, *shape-shifting* characters

- Malayalam script evolved from Brahmi, Grantha
- ~120 base characters encoded in Unicode
- ~800 conjuncts characters, formed from sequence of base characters
- Shape shifting
k+r+u: ക+ര+ു → ക്രു
- Vowel sign forms

900+, *shape-shifting* characters

- Malayalam script evolved from Brahmi, Grantha
- ~120 base characters encoded in Unicode
- ~800 conjuncts characters, formed from sequence of base characters
- Shape shifting

k+r+u: ക+ര+ു → ക്രു

- Vowel sign forms

k+u: ക+ു → കൂ

900+, *shape-shifting* characters

- Malayalam script evolved from Brahmi, Grantha
- ~120 base characters encoded in Unicode
- ~800 conjuncts characters, formed from sequence of base characters
- Shape shifting

k+r+u: ക+ര+ു → ക്രു

- Vowel sign forms

k+u: ക+ു → കൂ

c+u: ച+ു → ചൂ

900+, *shape-shifting* characters

- Malayalam script evolved from Brahmi, Grantha
- ~120 base characters encoded in Unicode
- ~800 conjuncts characters, formed from sequence of base characters
- Shape shifting

k+r+u: ക+ര+ു → ക്രു

- Vowel sign forms

k+u: ക+ു → കൂ

c+u: ച+ു → ചൂ

th+u: ത+ു → തൂ

900+, *shape-shifting* characters

- Malayalam script evolved from Brahmi, Grantha
- ~120 base characters encoded in Unicode
- ~800 conjuncts characters, formed from sequence of base characters
- Shape shifting

k+r+u: ക+ര+ു → ക്രു

- Vowel sign forms

k+u: ക+ു → കൂ

c+u: ച+ു → ചൂ

th+u: ത+ു → തൂ

n+u: ന+ു → നൂ

900+, *shape-shifting* characters

- Malayalam script evolved from Brahmi, Grantha
- ~120 base characters encoded in Unicode
- ~800 conjuncts characters, formed from sequence of base characters
- Shape shifting

k+r+u: ക+ര+ു → ക്ര

- Vowel sign forms

k+u: ക+ു → ക്ക

c+u: ച+ു → ച്ച

th+u: ത+ു → ത്ത

n+u: ന+ു → ന്ന

- Requires complex-text shaping support

METAFONT, METAPOST

Why METAFONT

We are mostly programmers, save a designer.¹

What can we do if we can't draw but program typefaces?

¹Gerard Unger “...but I don't want to become a programmer — let alone a parameterizer” [*Visible Language*, 16(4) (1982) 353–356]

Why METAFONT

We are mostly programmers, save a designer.¹

What can we do if we can't draw but program typefaces?

METAFONT, of course!

¹Gerard Unger “...but I don't want to become a programmer — let alone a parameterizer” [*Visible Language*, 16(4) (1982) 353–356]

Why METAFONT

We are mostly programmers, save a designer.¹

What can we do if we can't draw but program typefaces?

METAFONT, of course!


- Reusable shape components to create character shapes
- Parametric design: entire family (regular, bold, thin, slanted) from a **single** source
- Fine adjustments are easy thanks to programmability
- Distributed, version managed & collaborative development
- METAPOST → vector output (SVG)

¹Gerard Unger “...but I don't want to become a programmer — let alone a parameterizer” [*Visible Language*, 16(4) (1982) 353–356]

Components & shape library

Components



Reference font: RIT Rachana



| 3 3 3 3 3 3 3 3 3 3
3 3

Components

Reference font: RIT Rachana

	ഭ ഒ കെ രെ ണെ <u>കു</u> <u>ണ</u> <u>ഉ</u> <u>രു</u> <u>തു</u>
	ര രു നു ശു ണു കു

Components

Reference font: RIT Rachana



ഭ ഒ കെ തെ ഒ കു നു ഉ തു
തു



ര ര ത്ത ന്ത ശ്ത ണ്ത ക്ത



ത ത്ത ത്ത ത്ത ത്ത ത്ത ത്ത
തു ത്ത

... and about a 100 more components.

Shape library

```
%library of shape components
```

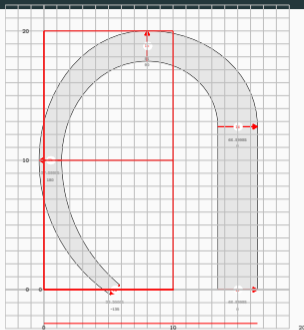
```
input ml-shape-lib;
```

```
beginfig(34);
```

```
%a component
```

```
coord_c_llobe (1) (0,0);
```

```
pstroke_c_llobe (1);
```



Shape library

```
%library of shape components
```

```
input ml-shape-lib;
```

```
beginfig(34);
```

```
%a component
```

```
coor_c_llobe (1) (0,0);
```

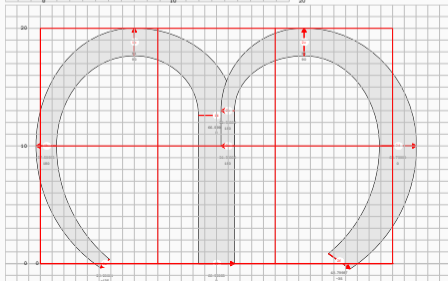
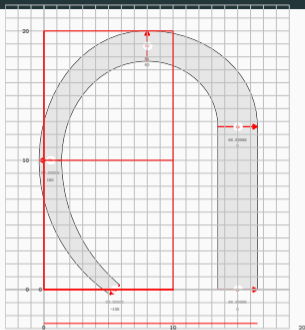
```
pstroke_c_llobe (1);
```

```
%another component
```

```
coor_c_rlobe (2) (x1f.r-.5wd2b,0);
```

```
pstroke_c_rlobe (2);
```

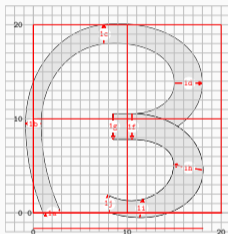
```
endfig;
```



Altering pre-defined shapes

Some character/component shapes need to be altered to be used in another character; such as moving a coordinate or changing width of penstroke at a coordinate.

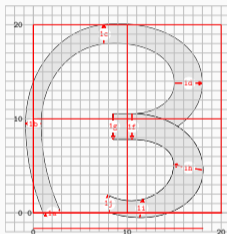
```
beginfig(32);  
% dx1b=-2u;  
coord_g_da (1) (0,0);  
pstroke_g_da (1);  
endfig;
```



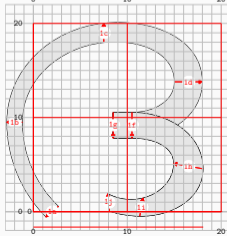
Altering pre-defined shapes

Some character/component shapes need to be altered to be used in another character; such as moving a coordinate or changing width of penstroke at a coordinate.

```
beginfig(32);  
% dx1b=-2u;  
coord_g_da (1) (0,0);  
pstroke_g_da (1);  
endfig;
```

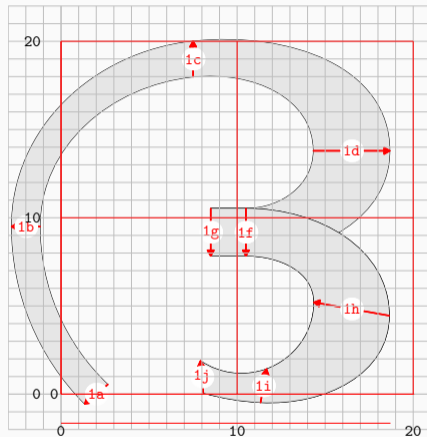


```
beginfig(32);  
dx1b=-2u;  
coord_g_da (1) (0,0);  
pstroke_g_da (1);  
endfig;
```



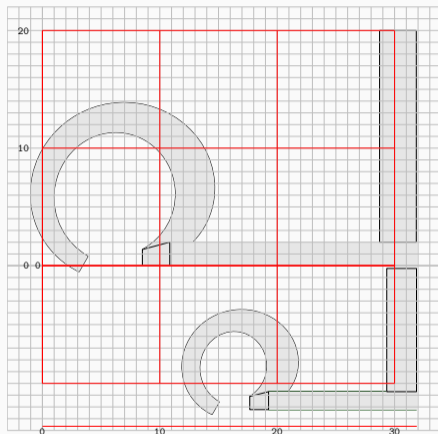
Change width and/or angle of penstroke.

```
beginfig(32);  
dx1b=-2u; % change x-pos  
coor_g_da (1) (0,0);  
ang1a=a_sw; % change angle  
wd1d=wd1h=w_ea+1t; % change width  
pstroke_g_da (1);  
endfig;
```

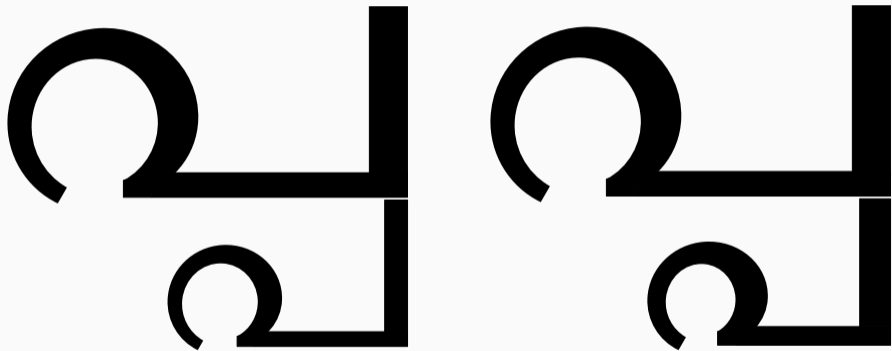


Stacked conjuncts.

```
beginfig(00);  
% first char with a prefix  
g1:=image(gl_pa(p));  
vconj:=true; width_angle(wa_n);  
% same char with different prefix  
g2:=image(gl_pa(pp));  
g3:=g2 xscaled .6 yscaled .6;  
currentpicture:=g1;  
addto currentpicture also g3  
    shifted (xpart (lrcorner g1)  
            -xpart(urcorner g3),  
            -(12u+5));  
endfig;
```



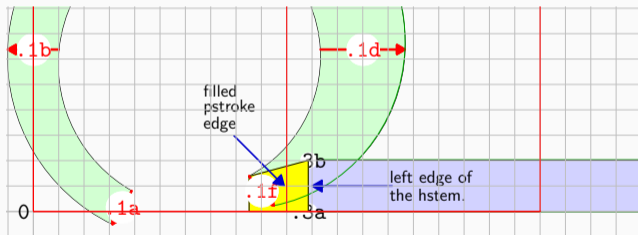
The vconj usage.



A bit of woodwork

When the left edge of a horizontal stem joins with the curved stroke of the character.

```
% definition:  
def pstroke_edge (expr ll,ul,ur,lr) =  
  filldraw ll -- ul -- ur -- lr --  
  cycle withcolor gcolor;  
enddef;  
% usage example:  
pstroke_edge (  
  ( x.prx.1f.l , y.prx.3a ),  
    z.prx.1f.l ,  
    z.prx.3b ,  
    z.prx.3a  
);
```



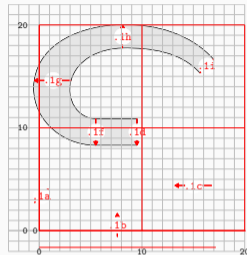
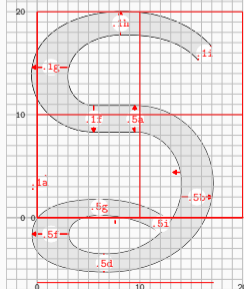
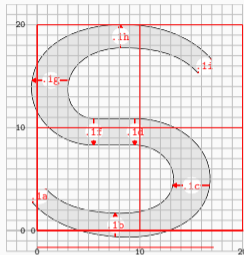
Reuse parts of a glyph

```
%% glyph def of t1 %%
```

```
def gL_Ta =  
  coor_g_ta (t1.1) (0,0);  
  % set start of subpath  
  reset_cut; start:=xstt;  
  % strokes from 'start'  
  pstroke_g_ta (t1.1);  
  reset_xst;  
endef;
```

```
%% glyph def of t1r1 %%
```

```
def gL_TR =  
  % draw curve from 4th point  
  xstt:=3; gL_Ta;  
  % attach bottom component  
  ang_cor:=-5;  
  coor_c_prkar (t1.5) (x.t1.1f,3u);  
  start:=0;  
  wd.t1.5a=wd.t1.1f;  
  x.t1.5f:=x.t1.1g;  
  x.t1.5a:=x.t1.1d;  
  y.t1.5a:=y.t1.1d;  
  wd.t1.5f:=wd.t1.1g;  
  pstroke_c_prkar (t1.5);  
  reset_cut;  
endef;
```



Vowel signs

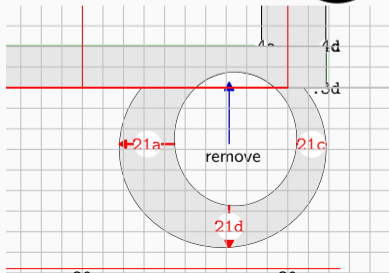
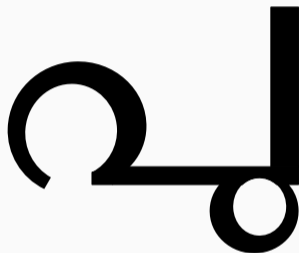
Vowel signs ു, ൂ, ൃ, ൄ conjoin with consonants to form conjuncts. The signs ു, ൂ (*u, ū*) themselves have 4 different forms depending on the base character.

ക, ര	<i>ku, ru</i>	ജ, റ	<i>jū, rū</i>
ഗ, ങ്ങ, ത	<i>gu, ju, thu</i>	ക, ഗ, ത	<i>kū, gū, thū</i>
ന, ണ, ന്ന	<i>nu, ṇu, nnu</i>	ന്റ, ണ്റ, ന്ന്റ	<i>nū, ṇū, nnū</i>
ദ, പ, ബ, മ	<i>du, pu, bu, mu</i>	ദ, പ, ബ, മ	<i>dū, pū, bū, mū</i>

Vowel signs

The vowel sign ‘incises’ into the base character. `unfil` doesn’t work with SVG, an ‘overlap removal’ is necessary.

```
%%% ɔ̄ %%%  
def gl_pu (suffix prx) =  
  outln:=true;  
  gl_pa(prx);  
  make_stem_u (21) (x.prx.3d-.5wd21c,  
    y.prx.3c-.75wd21b);  
enddef;  
%%% ɔ̄-sign %%%  
def make_stem_u (suffix $) (expr xsh,ysh) =  
  coor_vl_round_u_alt ($) (xsh,ysh);  
  pstroke_vl_round_u_alt ($);  
  if not noreverse: stem:= reverse stem; fi  
  find_outlines(rmpath,stem)(P);  
  for i=1 upto P.num: ypenstroke P[i]; endfor  
enddef;
```



OpenType font building

OpenType font

Binary font formats (OTF, TTF, WOFF2) are generated by a FontForge script, driven by config file.

METAPOST → SVG → FontForge + scripts → OTF/TTF/WOFF2

Config file:

```
#Metadata
```

```
[font]
```

```
family=Sayahna
```

```
name=Sayahna-Regular
```

```
version=0.9.1
```

```
ascent=820
```

```
descent=180
```

```
copyright=Copyright 2021-2023 Rachana
```

```
Institute of Typography
```

```
<info@rachana.org.in>
```

```
#SVG, Opentype feature file, Unicode mapping
```

```
[source]
```

```
glyphdir=svgs-regular/
```

```
featurefile=features/sayahna-feature.fea
```

```
ucglyphmapfile=tools/rit-ml-uc-glyph.map
```

```
scaleglyphs=True
```

```
#Width of specific glyphs
```

```
[width]
```

```
space=300
```

```
#Default and overridden left/right bearings
```

```
[bearing]
```

```
default=30,40
```

```
i1=-74,30 #negative left bearing
```

```
i2=-80,30
```

```
r1=-112,30
```

```
xx=-57,30
```

```
y2=-70,30
```

```
y2u1=-70,30
```

```
y2u2=-70,30
```

```
v2=-40,30
```


Thanks

Source code under free software licenses, LPPL and OFL:

<https://gitlab.com/rit-fonts/Sayahna-font/>

Acknowledgements

The METAFONT book

METATYPE1, plain_ex: JNS team

Jeroen Hellingman

Khaled Hosny

Feedback: rajeesh@rachana.org.in